

# Graph ODEs and Beyond: A Comprehensive Survey on Integrating Differential Equations with Graph Neural Networks

Zewen Liu\*

Emory University

[zewen.liu@emory.edu](mailto:zewen.liu@emory.edu)

Zijie Huang

Amazon

[zijiehjj@gmail.com](mailto:zijiehjj@gmail.com)

Xiaoda Wang\*

Emory University

[xiaoda.wang@emory.edu](mailto:xiaoda.wang@emory.edu)

Carl Yang

Emory University

[j.carlyang@emory.edu](mailto:j.carlyang@emory.edu)

Bohan Wang

Emory University

[bohan.wang2@emory.edu](mailto:bohan.wang2@emory.edu)

Wei Jin

Emory University

[wei.jin@emory.edu](mailto:wei.jin@emory.edu)

## Abstract

Graph Neural Networks (GNNs) and differential equations (DEs) are two rapidly advancing areas of research that have shown remarkable synergy in recent years. GNNs have emerged as powerful tools for learning on graph-structured data, while differential equations provide a principled framework for modeling continuous dynamics across time and space. The intersection of these fields has led to innovative approaches that leverage the strengths of both, enabling applications in physics-informed learning, spatiotemporal modeling, and scientific computing. This survey aims to provide a comprehensive overview of the burgeoning research at the intersection of GNNs and DEs. We will categorize existing methods, discuss their underlying principles, and highlight their applications across domains such as molecular modeling, traffic prediction, and epidemic spreading. Furthermore, we identify open challenges and outline future research directions to advance this interdisciplinary field. A comprehensive paper list is provided at <https://github.com/Emory-Melody/Awesome-Graph-NDEs>. This survey serves as a resource for researchers and practitioners seeking to understand and contribute to the fusion of GNNs and DEs.

## Keywords

Graph Neural Networks, Differential Equations, Deep Learning

## ACM Reference Format:

Zewen Liu, Xiaoda Wang, Bohan Wang, Zijie Huang, Carl Yang, and Wei Jin. 2025. Graph ODEs and Beyond: A Comprehensive Survey on Integrating Differential Equations with Graph Neural Networks. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3711896.3736559>

## 1 Introduction

Understanding and predicting complex behaviors in natural and engineered systems is a fundamental challenge across scientific and industrial domains. Many real-world phenomena exhibit dynamic evolution over time, governed by intricate interdependencies

\*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution 4.0 International License.  
KDD '25, Toronto, ON, Canada

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1454-2/2025/08

<https://doi.org/10.1145/3711896.3736559>

between variables. Examples include climate patterns shaped by atmospheric and oceanic interactions [1], population dynamics influenced by birth and migration rates [2], financial markets driven by investor behavior and economic indicators [3], disease progression driven by biological factors [4, 5], and the spread of infectious diseases determined by transmission dynamics and intervention strategies [6, 7]. Capturing these temporal changes and underlying mechanisms requires mathematical models that not only describe system behavior but also provide predictive insights.

To effectively model dynamical systems, Differential Equations (DEs), such as Ordinary Differential Equations (ODEs) [8], Partial Differential Equations (PDEs) [9], and Stochastic Differential Equations (SDEs) [10], relate one or more unknown functions to their derivatives, thus describing how outputs vary given changing variables. At their core, DEs consist of three essential components: (1) *state variables* that describe the system's condition, (2) *derivatives* that model and capture the rate of change, and (3) *parameters* that influence the dynamics under given initial and boundary conditions. These elements work together to provide a structured approach to understanding how systems evolve over time.

Despite the crucial role of DEs in modeling complex phenomena, various challenges arise from real-world applications. Notably, many systems exhibit intricate, high-dimensional dynamics that are difficult to capture using purely knowledge-driven DE formulations [11], as deriving accurate governing equations often requires human expert involvement. Moreover, computational efficiency remains a major obstacle, especially for high-dimensional and nonlinear PDEs, since traditional numerical solvers must manage an enormous number of equations corresponding to the system's graph structure, often rendering these approaches prohibitively expensive [12–14]. In response to these challenges, neural differential equations (NDEs), such as Neural ODEs [15], have emerged as a data-driven alternative that learns the underlying dynamics directly from data, bypassing the need for explicit formulation of governing rules. This innovative approach enables the modeling of systems for which traditional equations may be intractable or unknown. Nevertheless, while NDEs excel at capturing temporal evolution, it remains challenging to model spatial dynamics, such as epidemic spread in social networks [16] or transportation flows in urban networks [17], where discrete interactions complicate continuous-state representations. This limitation urgently calls for methods that can effectively integrate temporal dynamics with spatial context.

To handle the above issues, recent research has leveraged Graph Neural Networks (GNNs) [18–21], powerful tools for learning relational data, to build graph-based NDEs and model the complex interactions between variables. Early explorations integrate the graph learning capabilities of GNNs within the continuous-time framework of NDEs and propose Graph neural ODEs [22–24], which offer a versatile and powerful approach to modeling complex systems that evolve over both space and time. This integration not only enables the capture of dynamic temporal behavior but also leverages the rich spatial relationships encoded in graph structures. Beyond Graph Neural ODEs, the broader class of Graph Neural Differential Equations (Graph NDEs), including Graph Neural PDEs [25] and Graph Neural SDEs [26], bridges the gap between NDEs and GNNs.

**Contributions.** In this work, we aim to present a comprehensive and latest review of methods that combine graph neural networks with differential equations, addressing the gap by summarizing key tasks, methodologies, and applications in this evolving field. Our contributions can be summarized as follows:

- (a) We offer *the first comprehensive review* of Graph NDEs that model continuous spatial and temporal dynamics.
- (b) We introduce a structured taxonomy of Graph NDEs in Section 3 and conduct an in-depth review of research integrating GNNs with different classes of differential equations, including ODEs, PDEs, and SDEs, as detailed in Section 4.
- (c) We explore the diverse applications of Graph NDEs in Section 5, highlighting their impact across various real-world scenarios.
- (d) We identify emerging trends, key challenges, and promising future research directions in Section 6, aiming to inspire further exploration in this interdisciplinary field.

**Connections to existing surveys.** While previous surveys have explored Graph NDEs, they often lack comprehensiveness in methodology and categorization, limiting their ability to fully bridge GNNs and NDEs. Many focus on specific applications of neural differential equations [27–29], overlooking spatial dynamics. Others examine the integration of GNNs with differential equations [30, 31] but remain narrow in scope regarding DE types and categorization. In contrast, our survey compiles a broad range of recent studies, offering a detailed review of methodologies, challenges, and applications. Additionally, we present a well-structured taxonomy as well as valuable insights for future research.

## 2 Background

### 2.1 Learning on Graphs

In this paper, we define a graph as  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $|\mathcal{V}| = N$  represents the number of nodes, and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  represents the set of edges connecting nodes. The features of all nodes is represented as  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^{N \times D}$ , where  $D$  denotes the feature dimension. The adjacency matrix of  $\mathcal{G}$  is denoted as  $\mathbf{A}$ , where  $A_{ij} = 1$  if the edge  $e_{ij} \in \mathcal{E}$  and  $A_{ij} = 0$  if  $e_{ij} \notin \mathcal{E}$ . GNNs provide a flexible framework to learn graph representations. A common paradigm is message passing, where each node  $v$  updates its representation  $\mathbf{h}_v$  based on aggregating messages from its neighbors  $\mathcal{N}(v)$ . A GNN with  $L$  layers can be described as:

$$\mathbf{h}_v^{(l+1)} = f_\phi \left( \mathbf{h}_v^{(l)}, \bigoplus_{u \in \mathcal{N}(v)} f_\theta(\mathbf{h}_v^{(l)}, \mathbf{h}_u^{(l)}, \mathbf{e}_{uv}) \right), \forall l \in [L], \quad (1)$$

where  $f_\theta$  and  $f_\phi$  are learnable functions parameterized by  $\theta$  and  $\phi$ ,  $\mathbf{e}_{uv}$  denotes edge features (if available), and  $\bigoplus$  is permutation invariant aggregation operator that aggregates neighbor information. The final representation can then be used for downstream tasks such as link prediction and graph-level classification [32], etc.

### 2.2 Neural Differential Equations

Differential equations model dynamic systems across various domains, with their form varying based on the system. In the following, we illustrate three common types of DEs and NDEs.

**Ordinary Differential Equations (ODEs).** ODEs describe system evolution with respect to a single independent variable, typically time  $t$ . The general form is:  $\frac{dx}{dt} = f(x(t), t)$ , where  $x(t)$  is the system state, and  $f$  dictates its rate of change.

**Partial Differential Equations (PDEs).** PDEs involve multiple independent variables and their partial derivatives. A classical example is the diffusion equation, given by:  $\frac{\partial u}{\partial t} = \alpha \nabla^2 u$ , where  $u = u(x, t)$  represents the unknown quantity varying in spatial coordinate  $x$  and time  $t$ ,  $\alpha$  is the diffusion coefficient, quantifying the rate of spatial dispersion, and  $\nabla^2$  denotes the Laplacian operator, defined as the divergence of the gradient of the function  $u$ , capturing spatial changes in systems such as fluid dynamics [33].

**Stochastic Differential Equations (SDEs).** SDEs extend ODEs by modeling the evolution of a state variable  $x(t)$  through the incorporation of randomness, often via a Wiener process  $W_t$  [34]:

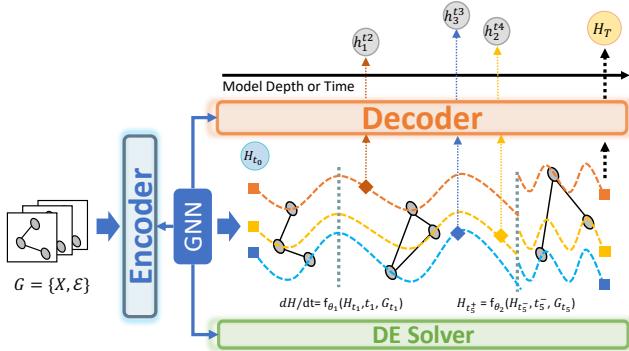
$$dx(t) = \mu(x(t), t)dt + \sigma(x(t), t)dW_t, \quad (2)$$

where  $\mu$  and  $\sigma$  are drift and diffusion terms. SDEs model systems with inherent randomness such as biological processes [35].

**Neural Differential Equations.** Neural Differential Equations extend classical differential equations by parameterizing the evolution function with neural networks. A prominent example is Neural Ordinary Differential Equations [15], where a neural network models the derivative of a latent state:  $\frac{dx}{dt} = f_\theta(x, t)$ , where  $f_\theta$  is a neural network parameterized by  $\theta$ . An ODE solver is used to compute the solution at any desired time point:  $x(t) = x(t_0) + \int_{t_0}^t f_\theta(x(\tau), \tau) d\tau$ , where  $t_0$  denotes the starting time point. For back-propagation, NODEs use the adjoint sensitivity method [36] to solve a second ODE backward in time to compute gradients efficiently:  $\frac{da}{dt} = -a^T \frac{\partial f_\theta}{\partial x}$ , where  $a = \frac{\partial L}{\partial x}$  is the adjoint state [15]. This approach enables training with constant memory cost.

### 2.3 Combining GNNs with DEs

By integrating the representational power of GNNs with the dynamic modeling capabilities of DEs, we introduce the concept of Graph Neural Differential Equations (Graph NDEs). A Graph NDE typically consists of two fundamental components: (i) a system of differential equations governing the temporal and spatial evolutions of states, parameterized by Neural Networks, and (ii) an initial condition that specifies the starting state of the system. While DEs establish the continuous or discrete dynamical progression of states, the role of GNNs is more flexible, as they can be incorporated at different stages of the modeling framework. As depicted in Figure 1, Graph NDEs can be roughly categorized based on the manner in which GNNs are embedded into the system dynamics.



**Figure 1: GNNs can function as an *encoder*, *decoder*, or *differential equation* in the Graph NDEs.** Firstly, the encoder maps inputs to a latent initial condition, which is then propagated by the DE solver over time or model depth. Furthermore, intermediate updates can modulate state evolution through state derivatives or direct reallocation. Finally, the decoder reconstructs the latent trajectory into the target space.

**2.3.1 Roles of GNNs.** Neural DEs generally operate in a latent space, where they model the evolution of states over time. Consequently, an encoding-decoding mechanism is typically employed: an encoder maps raw input data to a latent representation, and a decoder maps the evolved latent states into the target space. GNNs can be incorporated at various points in this pipeline, functioning as encoders and decoders and parameterizing the governing DEs. **GNNs as Encoders.** When GNNs function as encoders, they map node feature  $\mathbf{X} \in \mathbb{R}^{N \times D}$  into latent representations  $\mathbf{H} \in \mathbb{R}^{N \times D'}$  while preserving relational dependencies in the graph  $\mathcal{G}$  [37–40]. The encoding function can be expressed as:  $\mathbf{H} = \text{GNN}(\mathbf{X}, \mathcal{G}; \Theta_{enc})$ , where  $\Theta_{enc}$  denotes the parameters of the GNN encoder. Related works usually incorporate spatial-temporal GNNs [41, 42], capturing both the structural and temporal information.

**GNNs as Decoders.** Given a latent representations  $\mathbf{H}(t)$  evolved over time using DEs, a decoder maps it to the target space via:  $\hat{\mathbf{Y}} = \text{GNN}(\mathbf{H}(t), \mathcal{G}; \Theta_{dec})$ , where  $\Theta_{dec}$  is the decoder parameter. Graph-based decoders enable mapping that preserves node interactions and adapts to changes to graph topology [43–46].

**GNNs as Differential Equations.** Beyond serving as encoders or decoders, GNNs can be directly embedded within the differential equation to govern continuous state flows [22, 47–50]. Let  $x(t)$  denote the state of nodes at time  $t$ . The dynamics of  $x(t)$  can be described by a GNN-parameterized differential equation, taking Graph Neural ODE as an example:  $\frac{dx(t)}{dt} = \text{GNN}(x(t), \mathcal{G}, \Theta)$ , where  $\Theta$  is the parameter of the GNN that models continuous aggregation across the graph. Similarly, higher-order formulations extend this principle as:  $\frac{d^k x(t)}{dt^k} = \text{GNN}(x(t), \mathcal{G}, \Theta)$ , where  $k$  denotes the order of the DE. Such formulation enables enhanced expressiveness and adaptability in evolving graph structures [13, 51–58].

**2.3.2 Initial Condition Construction.** The definition of the initial condition  $x(t_0)$  significantly influences the trajectory of the learned dynamics. Besides latent encodings, initial states can be derived from raw inputs or learned embeddings.

**Encoding-Based Initialization.** In this case, raw node features  $\mathbf{X}$  are first mapped into a latent space before being used as the latent

initial condition, which can be deterministic:  $\mathbf{H}(t_0) = f(\mathbf{X}, \mathcal{G}; \Theta_{enc})$  or sampled from a given distribution [41, 42, 48, 59] (e.g. Gaussian):  $\mathbf{H}(t_0) \sim \mathcal{N}(\hat{\mu}, \hat{\sigma})$ , where  $\hat{\mu}$  and  $\hat{\sigma}$  are inferred by a function  $f(\mathbf{X}, \mathcal{G})$ , introducing stochasticity into the initial states to model uncertainty.

**Pre-defined Initialization.** In physical systems, the initial condition is often dictated by domain constraints, leading to  $\mathbf{H}(t_0) = \mathbf{X}$ , where no additional encoding is applied. This approach is common in dynamic simulations [60–62], where initial states are predefined or randomly generated.

**Learning-Based Initialization.** Instead of explicitly defining initial states and encoding raw features, it is also viable to learn the initial condition during model training. The model learns an optimal embedding  $\mathbf{H}(t_0)$  that best facilitates downstream tasks:  $\mathbf{H}(t_0) = \operatorname{argmin}_{\mathbf{H}(t_0)} \mathcal{L}(\mathcal{F}(\mathbf{H}(t_0), \mathcal{G}))$ , where  $\mathcal{L}$  represents a task-specific objective function and  $\mathcal{F}$  denotes the differential equation dynamics. This strategy is particularly effective in recommendation systems [63–65], where graph structure alone is available, and node representations must be inferred from relational interactions.

### 3 Taxonomies

We classify Graph NDEs based on tasks, datasets, graph construction techniques, and methodological distinctions. A complete categorization is in Appendix A, with a partial version in Figure 2.

#### 3.1 Tasks

Among all the research investigated in this paper, Graph NDEs are applied in five primary tasks: *Node/Graph Classification*, *Link Prediction*, *Ranking*, *Forecasting*, and *Graph Generation*.

**Node/Graph Classification.** Given  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , the goal is to learn functions  $f : \mathcal{V} \rightarrow \mathcal{Y}$  (node classification) or  $g : \mathcal{G} \rightarrow \mathcal{Y}$  (graph classification). Graph NDEs model message passing as a continuous process rather than a one-step discrete propagation. For citation network prediction [66], articles (nodes) and citations (edges) form citation graphs, where Graph NDEs improve representation learning by capturing continuous citing patterns.

**Link Prediction.** Given  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , link prediction learns  $h : \mathcal{V} \times \mathcal{V} \rightarrow [0, 1]$  to estimate edge existence probability. Graph NDEs enhance prediction by modeling continuous node embedding dynamics. Such a task is commonly seen in recommendation systems and knowledge graphs [58, 67].

**Ranking.** Ranking assigns scores to nodes, optimizing  $s : \mathcal{V} \rightarrow \mathbb{R}$  for ordered retrieval. Graph NDEs leverage continuous diffusion to model information propagation. For example, in recommendation systems [39], users and items form bipartite graphs; Graph NDEs capture evolving preferences, refining interaction prediction.

**Forecasting.** For dynamic graphs  $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$  with temporal node features  $\mathbf{X}(t)$ , forecasting estimates future states via  $f : \mathcal{V} \times \mathbb{R}^s \rightarrow \mathbb{R}^z$ , where  $s$  denotes the history input size and  $z$  refers to the horizon of prediction. Graph NDEs, incorporating time-continuous dynamics, excel in capturing gradual state transitions. For example, in traffic flow forecasting [50, 68–70], road networks use Graph NDEs for real-time flow updates and long-term forecasting.

**Graph Generation.** Given a training set  $\mathcal{G}_i$ , graph generation models  $p(\mathcal{G})$  to sample structurally meaningful graphs. Graph NDEs enable continuous latent space exploration, improving structural coherence and diversity in generated samples [71, 72].

### 3.2 Graph Construction

The construction of graphs plays a crucial role in shaping both the design and performance of Graph NDEs. This process can be analyzed along two key dimensions: *spatial* and *temporal*.

**3.2.1 Spatial Level.** In this paper, we depict *spatial-level relations* as a general notion of proximity or relationships between nodes. Depending on how nodes and edges are formulated, graph construction typically follows one of two primary approaches:

**Point-Based Graphs.** Point-based graphs can be irregular, where nodes correspond to individual data points, and edges are established based on a function  $f_e: \mathcal{V} \times \mathcal{V} \rightarrow \{0, 1\}$  that determines connectivity based on proximity from observations. The notion of proximity can be defined in various ways depending on the nature of the data, often reflecting spatial [73], or semantic relationships [74] among nodes. To quantify the strength of connectivity, edge weights can also be applied.

**Grid-Based Graphs.** A grid-based graph is a spatially regular, where  $\mathcal{V} \subset \mathbb{Z}^n$  represents nodes positioned at integer lattice points in an  $n$ -dimensional space, and edges  $\mathcal{E}$  connect nodes based on a predefined neighborhood structure. This structured representation is widely used in drone swarming [75], and physical modeling [12–14, 51], where data is arranged in a spatially regular manner.

**3.2.2 Temporal Level.** Temporal graphs evolve over time in terms of node features or graph topology. Therefore, each time point yields a distinct graph  $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$ . Formally, temporal graphs can be categorized into:

**Static Graph.** The temporal evolution of a static graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is solely captured through time-dependent node attributes  $X(t)$ . That is, each node  $v \in \mathcal{V}$  has a feature vector  $x_v(t)$  evolving over time, while the edge set remains unchanged, i.e.,  $\mathcal{E}(t) = \mathcal{E}, \forall t$ . Applications span citation [22] and traffic [70] networks.

**Dynamic Graph.** A dynamic graph is characterized by time-evolving edges and edge weights,  $\mathcal{E}(t)$ , meaning both connectivity and interaction strengths change over time. The structure of such graphs is determined via: i) explicit modification of adjacency relations, where  $\mathcal{E}(t) = \{e_{uv}(t)\}$  updates based on new inputs; ii) adaptive learning of edges and weights during training, where adjacency matrices are replaced by a learned attention matrix  $A_t = (a_{ij}) \in \mathbb{R}^{N \times N}$ , with  $a_{ij} = f(v_i, v_j, t)$  capturing dynamic influence. The temporal evolution of node states follows Graph NDEs:  $\frac{dx(t)}{dt} = f_\theta(x(t), \mathcal{E}(t))$ , which model node state evolution under varying graph structure. Applications span social interactions [37], graph generation [71, 72], etc.

### 3.3 Modeling Spatial & Temporal Dynamics

Graph NDEs surpass discrete models by providing a continuous flow of latent states across spatial and temporal dimensions, and the key lies in the modeling of *spatial* and *temporal dynamics*.

**Temporal Dynamics Modeling.** In classical NDEs, the continuous evolution of variable states is defined with respect to actual time,  $t \in \mathbb{R}^+$ , ensuring alignment with the target trajectory [15]. Specifically, in Graph NDEs, the state of each node,  $x(t)$ , evolves according to a time-dependent differential equation  $\frac{dx(t)}{dt} = f_\theta(x(t), t)$ , where the graph structure is either encoded in the initial condition  $x(t_0)$  [38] or integrated into the function  $f_\theta$ . Additionally, temporal

dynamics are influenced by external controls or new inputs, which can update node states and introduce new graph structures, thereby altering the flow of states over time. Spatial-temporal models extend NDEs by explicitly incorporating the spatial dimension, capturing both spatial and temporal evolution on dynamic graphs.

**Spatial Dynamics Modeling.** The dynamic spatial evolution can also be applied to static graphs, where the model *depth* [23] corresponds to a continuous notion of time. Unlike the conventional approach of stacking discrete GNN layers, this continuous perspective naturally connects to diffusion equations [76]. A general graph diffusion equation is given by:

$$\frac{\partial x(t)}{\partial t} = \text{div}\left(G(x(t), t) \cdot \nabla x(t)\right), \quad (3)$$

where  $\nabla x(t)$  is the divergence of  $x$ ,  $G(x(t), t)$  is the diffusion coefficient that may depend on both the current state  $x(t)$  as well as current time or depth  $t$ , and  $\text{div}(\cdot)$  denotes the graph divergence operator. By parameterizing this diffusion process, we arrive at the formulation of NDEs on graphs.

Recent work provides further insight into this connection. For instance, Chamberlain et al. [77] show that GNNs can be viewed as the discrete form of Beltrami flow, while Choi et al. [78] develop a reaction-diffusion-based GNN architecture. Taken together, these studies reinforce the link between Graph NDEs and graph diffusion processes. Consequently, Graph NDEs provide both smoother feature propagation across the graph and a principled physical analogy grounded in a well-established diffusion process.

## 4 Methodology

In this section, we detail the methodology underpinning Graph Neural Differential Equations (Graph NDEs). Our discussion is organized around two primary perspectives: *Temporal Dynamics Modeling* and *Spatial Dynamics Modeling*. For each perspective, we elaborate the unique challenges involved.

### 4.1 Temporal Dynamics Modeling

For spatial-temporal models, incorporating the time dimension presents several challenges. This section outlines key temporal modeling challenges and corresponding solutions in Graph NDEs, including *dynamic updates*, *irregular time intervals*, *modeling temporal delay*, *modeling hybrid system dynamics*, and the *efficiency*.

**4.1.1 Dynamic Temporal Updates.** For a naive DE, whether parameterized by an NN or not, a fixed trajectory is predicted once the initial condition is given. This is because the DE defines the evolution or flow of node states, namely the vector field [15]. However, in a spatial-temporal graph, both node features and the graph structure can evolve over time independently of the vector field defined by the DE. These updates significantly impact the target trajectory. From the perspective of the vector field, the updated graphs can be interpreted in two ways: 1. Reallocation of states, and 2. Conditioned states flow. For the reallocation of states, the vector field does not change while the new inputs introduce a jump of states in the vector field:  $x(t_k^+) = \Phi(x(t_k^-), \Delta(t_k))$ , where  $t_k$  denotes the moment of a discrete update,  $t_k^-$  and  $t_k^+$  are the times just before and just after the jump, respectively, and  $\Delta(t_k)$  captures the new information or structural change at  $t_k$ . The update function  $\Phi$  then

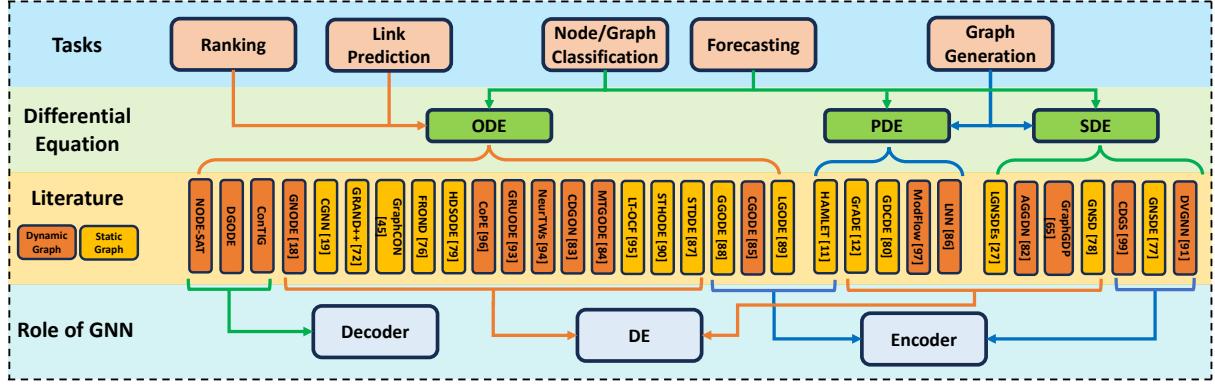


Figure 2: Summary of Graph Neural Differential Equation methods.

adjusts the node states accordingly. This jump effectively resets the trajectory of the node states, altering the dynamics governed by the DE. For example, Poli et al. [22] introduced an autoregressive graph differential equation that applies "jumps" to adapt to the dynamic graph structure., and Zhang et al. [79] applied Graph NDE in the recommendation system, where jumps of node features are introduced as the interaction between the user and item changes.

In the case of a conditioned state flow, newly arrived inputs at a given time point serve as the input of Graph NDEs, which gives a formulation of  $\frac{d\mathbf{x}}{dt} = F(\mathbf{X}(t), \mathcal{G}(t))$ , where  $\mathbf{X}(t)$  is the current node states and  $\mathcal{G}(t)$  is the dynamic graph including node features and the graph structure. The change of  $\mathcal{G}(t)$  can be viewed as either a change of the vector field or as an adjustment to the output  $\frac{d\mathbf{x}}{dt}$ . Instead of solely relying on the initial condition, the vector field becomes conditioned on the current input, allowing the system to adapt its trajectory in response to external influences [80, 81].

**4.1.2 Irregular Time Interval.** Real-world dynamic systems exhibit irregularly sampled time series, where observations occur at non-uniform intervals, making traditional discrete-time models ineffective. DE-based models naturally address this by modeling the continuous evolution of node states so that the states at arbitrary time points can be inferred. Recent advancements, such as LG-ODE [82], CG-ODE [83], and GG-ODE [84], extend such ability to graph-structured data. LG-ODE [82] models continuous node dynamics using latent ODEs, enabling interpolation across uneven time steps. CG-ODE [83] further generalizes this by incorporating evolving graph structures, where both node states and edge interactions are learned through coupled differential equations. GG-ODE [84] extends these ideas across multiple environments by introducing environment-specific latent factors, enabling the transfer of learned dynamics across different systems. These approaches provide a flexible framework for modeling real-world graph dynamics under irregular sampling, outperforming discrete-time methods in handling asynchronous, partially observed data.

**4.1.3 Temporal Delay Modeling.** Traditional GNNs assume immediate information propagation, which fails to capture the inherent temporal delays present in real-world systems. In applications such as traffic forecasting, changes in one location take time to influence others, making delay-aware modeling essential. Long et al. [85] introduces the Spatial-Temporal Delay Differential

Equation, which explicitly incorporates time delays into spatial-temporal modeling. The core idea is to model node interactions as:  $\frac{dh_i(t)}{dt} = f(h_i(t), h_j(t - \tau_{ij}), \theta)$ , where  $\tau_{ij}$  represents the time delay in information propagation between nodes. Instead of assuming fixed delays, they propose two approaches: (1) a precomputed delay estimator using *max-cross correlation*, and (2) a time-delay estimator that dynamically adjusts  $\tau_{ij}$  based on traffic conditions.

**4.1.4 Temporal Dynamics for Hybrid Systems.** While Graph NDEs parameterize the DEs with NNs or GNNs, making the model inherently data-driven, incorporating domain-specific biases can be beneficial for regularizing the model outputs. Such modeling biases, often derived from well-established physical laws, constraints, or expert knowledge, are fundamental in traditional knowledge-driven models. To enhance the effectiveness and interpretability of Graph NDEs, hybrid models [7] integrate domain-knowledge into the modeling of Graph NDEs by either *constraining the form of Graph NDEs* or *predicting the abstract quantities or parameters of the physical system*. Li et al. [86] extend the use of the spatial-temporal decay model from one-dimensional dynamics to the high-dimensional latent space. Similarly, Han et al. [87] make use of the structure of the Susceptible-Infected-Recovered model and switch the modeling to the latent space. On the other hand, Sanchez-Gonzalez et al. [88] model the Hamiltonian mechanism by predicting the momentum and velocity in the physical function using a GNN. Similar practices have been made for Lagrangian mechanism [89].

**4.1.5 Efficient Temporal Simulation.** Although classic PDE models effectively describe various real-world phenomena with numerical solvers such as the Finite Element Method [90], time complexity remains a significant challenge, particularly for solving complex dynamics and real-time processing tasks [91]. The primary sources of this challenge are twofold. First, the high dimensionality and non-linearity of many problems lead to more intricate PDE systems, both in terms of the number of equations and their structural complexity. Second, the demand for larger or more fine-grained grid-based graphs further increases the number of nodes to be modeled [92], which can be computationally expensive using classic PDE solvers. Beyond the computational inefficiency of handling large-scale and complex PDEs, additional challenges arise when dealing with problems that lack a fixed PDE formulation. To address these issues, Graph Neural PDEs [12–14, 51] employ a data-driven approach to learn and integrate the governing rules of both spatial

and temporal dynamics across all nodes using a single model. Unlike traditional methods, where computational complexity scales with problem intricacy and the graph size, Graph Neural PDEs maintain a fixed model for all cases and does not rely on classic PDE solvers. As a result, the time complexity does not increase with the problem complexity, and the governing dynamics are encapsulated within the learned parameters, enabling efficient and scalable solutions.

## 4.2 Spatial Dynamics Modeling

Unlike spatial-temporal dynamics which model latent state evolution over time, spatial dynamics can also be modeled in terms of model depth. While traditional GNNs effectively capture static relationships through local aggregation, their discrete nature constrains their ability to represent continuous feature evolution. Here, we show how embedding GNNs in differential equations enables dynamic graph modeling and overcomes classic challenges such as *over-smoothing*, *measuring uncertainty*, *adversarial robustness*, *graph heterophily*, and *modeling high-order relations*.

**4.2.1 Over-Smoothing on Graphs.** GNNs with deeper architecture experience severe performance degradation due to vanishing gradient and over-smoothing, where node representations become indistinguishable and converge to the same value as more layers are added [93]. To mitigate the effect of over-smoothing, previous studies imitate residual networks [94] and develop skip connections [95] between layers, updating the node features using  $\mathbf{H}(t+1) = \mathbf{H}(t) + \text{GNN}_\theta(\mathbf{H}(t), \mathcal{G})$ , where  $t$  denotes the layer index. On the other hand, Graph NODE [22] takes a step further by making this process differential:  $\frac{d\mathbf{H}(t)}{dt} = \text{GNN}(\mathbf{H}(t), \mathcal{G})$ . Then, a numerical solver [96] is applied to acquire the trajectory. Additionally, CGNN [23] introduces the initial latent embeddings,  $E = \mathbf{H}(t_0)$ , into the ODE formulation:  $\frac{d\mathbf{H}(t)}{dt} = \text{GNN}(\mathbf{H}(t), \mathcal{G}) + \mathbf{H}(t_0)$ . This results in a function with a restart distribution, which helps the model retain the initial representations and effectively mitigates over-smoothing. Moreover, GRAND++ [97] interprets over-smoothing through the lens of diffusion, where deeper networks excessively diffuse node features, ultimately leading to uniform feature representations across all nodes. Similar to CGNN, Grand++ mitigates this issue by introducing a source term (restart) in the differential equation (DE) to preserve initial representations. Likewise, Graph-Coupled Oscillator Networks [52] establish a connection between over-smoothing and zero-Dirichlet energy steady states, proposing a second-order ODE to counteract the over-smoothing. Furthermore, Maskey et al. [98] extend the problem to directed graphs and tackle over-smoothing with fractional graph Laplacians.

**4.2.2 Uncertainty within Graph Dynamics.** Real-world data often contains noise and unobserved external factors that influence the dynamics of graph propagation. For one thing, graphs are often constructed from real-world data where the connections (edges) between nodes can be incomplete, noisy, or even spurious [99]. For another thing, even though a clear graph structure is given, the way information, influence, or any form of signal diffuses through these networks can be highly variable and subject to external factors (e.g., weather, human behavior) [100]. Since GNNs and Graph NODEs make predictions conditioned on neighbors, both kinds of uncertainty impact their performance. To provide a direct measurement

of uncertainty and improve the robustness of these models, Graph Neural SDEs introduce a stochastic diffusion term  $\sigma(x(t), t)dW_t$ , as illustrated in Equation 2, enhancing the performance on node classification tasks in both the In Distribution and Out of Distribution cases [34, 101, 102]. Furthermore, Liang et al. [103] combines graph variational encoding with SDE, generating dynamic graphs for spatial-temporal forecasting. Xing et al. [80] stacked the SDE module upon ODE, which works as a control signal to modulate the SDE propagation. Huang et al. [72] applied SDE in the graph generation task by applying the reverse-time SDE to generate the target permutation-invariant graphs from random graphs.

**4.2.3 Graph Adversarial Robustness.** GNNs are vulnerable to adversarial perturbations due to inter-node information exchange. Adversaries can perform modification attacks by adding or removing edges or injection attacks by introducing malicious nodes. Song et al. [104] treat graphs as discretized Riemannian manifolds and analyze the stability of the heat kernel under metric perturbations. Their results show that for small perturbations  $\varepsilon = o(1)$ , the change in node features remains bounded  $\|\varphi(u, t) - \hat{\varphi}(u, t)\| = O(\varepsilon)$ , where  $\varphi(u, t)$  is the node attribute of node  $u$  at time  $t$ , indicating that PDE-based GNNs can better withstand adversarial topology attacks. Building on this analysis, they propose a novel class of graph neural PDEs with stronger defenses against such adversarial modifications. While Song et al. [104] demonstrate Lyapunov stability, it does not necessarily guarantee adversarial robustness. Zhao et al. [105] analyze various stability concepts for graph neural flows, leading to the Hamiltonian Graph diffusion class, which improves robustness by maintaining constant total Hamiltonian energy over time, ensuring bounded BIBO stability. Recently, Kang et al. [106] made an extension to graph neural fractional-order differential equations, showing more robust than existing Graph neural ODEs.

**4.2.4 Graph Heterophily.** GNNs have been widely used for various graph-based learning tasks, yet they often assume connected nodes have similar attributes (homophily), which is not hold in heterophilic graphs, leading to suboptimal performance. To address this, recent works have explored Neural ODEs combined with graph dynamic modeling to enhance node representation learning in heterophilic settings. Recent studies have introduced diffusion-based models to handle heterophilic graphs effectively. Zhao et al. [107] propose a Graph Neural Convection-Diffusion framework, leveraging the convection-diffusion equation (CDE) to incorporate both homophilic and heterophilic information. The convection-diffusion equation is formulated as:  $\frac{\partial x}{\partial t} = \text{div}(D\nabla x) - \text{div}(vx)$ , where the first term represents diffusion, and the second term accounts for convection with velocity field  $v$  controlling the propagation direction. In the discrete graph setting, this extends to:  $\frac{\partial x(t)}{\partial t} = \text{div}(D(x(t), t) \odot \nabla x(t)) + \text{div}(v(t) \circ x(t))$ , where  $v(t)$  now adapts to node dissimilarity, enhancing classification performance on heterophilic graphs. Similarly, Zhang and Li [108] introduce a dual-channel Continuous Graph Neural Network with latent states applied using low-pass ( $\mathbf{H}_L$ ) and high-pass ( $\mathbf{H}_H$ ) filtering :

$$\frac{\partial \mathbf{H}_L}{\partial t} = (\hat{\mathbf{A}}_{\text{sym}} - \mathbf{I})\mathbf{H}_L + \mathbf{H}(t_0), \quad \frac{\partial \mathbf{H}_H}{\partial t} = (-\hat{\mathbf{A}}_{\text{sym}})\mathbf{H}_H + \mathbf{H}(t_0). \quad (4)$$

where  $\hat{\mathbf{A}}_{\text{sym}}$  is the symmetrically normalized adjacency matrix, and the features are mixed from both channels in the end.

**4.2.5 Graph Dynamics with High-Order Relations.** Many real-world problems involve interactions that go beyond pairwise relationships. Hypergraph learning [109] addresses this by allowing each hyperedge to connect multiple nodes. Nevertheless, incorporating hypergraph learning into the framework of Graph NDEs is non-trivial as the dynamics of pair-wise graphs and hypergraphs are different. To bridge the gap, Yao et al. [110] propose to model spatial and temporal evolutions separately, building a spatial hypergraph  $G_{sp}$  and a temporal hypergraph  $G_{te}$ . Then, hypergraph convolution is integrated into the ODE, which yields the spatial and temporal evolutions on the two graphs. In the end, an MLP layer is applied to combine the final embeddings from the spatial and temporal levels. Besides separately encoding the spatial and temporal evolutions, Yan et al. [111] propose to encode the node embeddings  $\mathbf{H}_v$  and hyperedge embeddings  $\mathbf{H}_e$  separately, which gives an ODE in the form of:  $\begin{bmatrix} \dot{\mathbf{H}}_v \\ \dot{\mathbf{H}}_e \end{bmatrix} = \begin{bmatrix} g_v(\mathbf{H}_v(t)) \\ g_e(\mathbf{H}_e(t)) \end{bmatrix} + A \begin{bmatrix} \mathbf{H}_v(t) \\ \mathbf{H}_e(t) \end{bmatrix}$ , where  $g_v$  and  $g_e$  are the control functions and  $A$  denotes the diffusion velocity effect between the vertex representation and the hyperedge representation in the dynamic system by the correlation of the hypergraph.

## 5 Applications

Graph NDEs have been applied across various domains due to their capability to model continuous spatial and temporal dynamics. In this section, we discuss on some popular applications, including *Physics Systems Simulation*, *Traffic Flow Forecasting*, *Recommendation Systems*, *Epidemic Modeling*, and *Graph Generation*.

### 5.1 Physics Systems Simulation

Graph Neural ODEs have proven effective for modeling continuous-time dynamics in physics-based simulations by parameterizing system evolution through differential equations, allowing flexible and efficient trajectory prediction. These models are particularly useful for simulating multi-body interactions [84], particle dynamics [112], spring system [48], charged particle system [48], chaotic pendulum system [113] and fluid mechanics [114], where relational structures naturally fit graph representations. For example, the Hamiltonian Graph Network [115] incorporates Hamiltonian mechanics to enforce energy conservation in learned physics models. GG-ODE [84] introduces environment-specific latent factors to adapt physics models across different conditions. EGODE [62] extends Graph Neural ODEs to hybrid systems, handling sudden state changes, e.g., rigid-body collisions. Furthermore, GNSTODE [116] improves spatial-temporal modeling in physics systems by learning latent force interactions and refining long-range dependencies. These models demonstrate how Graph ODEs enhance the accuracy, efficiency, and generalization of physics-based simulations, outperforming numerical solvers in long-term stability and adaptability.

### 5.2 Traffic Flow Forecasting

Traffic flow forecasting is a crucial task in intelligent transportation systems, requiring models that can capture complex spatial-temporal dependencies. Traditional models, such as ARIMA and LSTM-based approaches, struggle with irregular traffic patterns and

evolving road network dynamics. Graph NDEs provide a continuous-time framework that integrates spatial-temporal dynamics, improving long-range forecasting and adapting to variable time intervals. For example, STGODE [117] models traffic flow as a continuous dynamical system, integrating GNNs with an ODE solver to handle long-term dependencies. GODE-RNN [118] combines Graph NDEs with RNNs, capturing both fine-grained temporal changes and spatial interactions. ASTGODE [119] introduces an attention mechanism within Graph ODEs to enhance interpretability and adaptive forecasting. Additionally, GRAM-ODE [120] employs multiple Graph ODE modules to learn hierarchical traffic patterns, while AGODE [121] dynamically updates the graph structure to reflect changing traffic conditions. These methods outperform discrete-time GNN models in accuracy, demonstrating the potential of Graph NDEs to handle irregular and evolving traffic data efficiently.

### 5.3 Recommendation Systems

Recommendation systems [122] naturally form a bipartite graph structure, capturing relationships between users and items. Graph NDEs effectively model the continuous evolution of user preferences, surpassing traditional collaborative filtering methods [123] by accounting for dynamic interactions. For example, Qin et al. [124] propose an autoregressive propagation framework with an edge-evolving mechanism and a temporal aggregation module to predict user-item interactions, which is similar to CoPE [79] and ConTIG [46]. To further enhance the learned representations of Graph NDEs, Yang et al. [125] integrate contrastive learning into their optimization process. Additionally, to improve adaptability to dynamic graphs, Guo et al. [39] propose t-Alignment, which synchronizes the updating time steps of temporal session graphs within a batch.

### 5.4 Epidemic Modeling

Modeling infectious disease spread [7, 126, 127] is crucial for public health policy design. In an epidemic graph, nodes represent individuals or communities, while edges denote their interactions. Graph NDEs extend beyond the traditional mechanistic models, e.g., SIR and its variants [128], introducing *hybrid models* [7] that integrate both GNNs and mechanistic models. These approaches enhance the ability to capture complex infection dynamics. For example, STAN [129] preserves the ODE function of the SIR model and uses GNNs to predict the parameters of the SIR model on a static graph. As an extension, MepoGNN [130] adopts a graph learning module, which introduces learning on dynamic graphs. Besides learning the parameters of mechanistic models, Wan et al. [131] integrate the ODE function of the SIR model with GNNs, modeling the variables in the high-dimensional latent space.

### 5.5 Graph Generation

Graph generation is essential for applications like drug discovery and program synthesis [132], but modeling graph distributions is challenging due to their discrete, permutation-invariant nature. Traditional models like variational autoencoders [133] struggle with this invariance. Score-based generative models address this by using graph SDEs to simulate graph trajectories, where diffusion corrupts graphs into a prior distribution (e.g., normal distribution), and the trajectory captures both diffusion and denoising. These models rely

on log-density gradient vector fields, imposing fewer constraints than likelihood-based approaches while ensuring permutation invariance. The edge-wise dense prediction GNN [134] estimates scores for graph distributions while maintaining permutation invariance but is limited to adjacency matrices. To overcome this, Jo et al. [135] proposed Graph Diffusion via SDEs, which models both node features and adjacency matrices with separate drift and diffusion terms, capturing node-edge dependencies. CDGS [136] further improves this with hybrid message-passing blocks and fast ODE solvers, enabling rapid, high-quality molecule generation.

## 6 Future Work

While significant advancements in Graph NDEs, many challenges remain largely unexplored. In this section, we discuss these issues and suggest directions for future research.

### 6.1 Discovering Graph Differential Equations

Equation discovery is an essential task across scientific disciplines, facilitating the extraction of explicit mathematical relationships directly from observed data [137]. At the heart of this process lies *symbolic regression*. Recently, deep learning-based methods, such as set-to-sequence transformers [138] and large language models [139], have emerged as viable alternatives to traditional symbolic regression approaches. However, due to inherent architectural constraints, they focus on discrete representations of data. In contrast, Graph NDEs explicitly capture continuous dynamical systems by learning vector field representations, yet they typically encode equations in implicit forms. This fundamental difference indicates that integrating discrete-focused transformer-based methods with continuous-based Graph NDEs presents a compelling pathway toward advancing the field of differential equation discovery.

### 6.2 Handling Graph Sparsity and Sporadicity

Data sparsity in dynamic systems remains a critical challenge, often manifesting as limited labeled nodes and missing or incomplete observations over time and space. Additionally, sporadic patterns, characterized by both sparsity and irregular, unpredictable distribution, further challenge learning and inference [140]. Recently, Luo et al. [141] combined the strengths of neural processes and neural ODEs to model evolving graphs with missing edges and to capture physical dynamics from highly sparse spatial-temporal data. Nevertheless, open challenges persist in ensuring the robustness and efficiency of Graph NDEs under extreme sparsity conditions. To further address the challenge, possible solutions may involve zero-shot annotator to label a small portion of nodes [142], or graph condensation [143] that yields a condensed graph from sparse graph.

### 6.3 Scalability on Large Dynamic Graphs

Scalability remains a significant challenge for Graph NDEs, particularly when applied to large-scale dynamic graphs. These models require the solving of continuous-time differential equations for potentially millions of nodes and edges, leading to high computational overhead and memory demands [144]. The iterative nature of DE solvers exacerbates the issue since repeated evaluations of neural network functions over numerous time steps can be prohibitively time-consuming. To tackle this challenge, several approaches

can be explored: developing efficient numerical solvers tailored for neural differential equations [145], leveraging parallelization and GPU acceleration [146], and employing sparse representations and approximation methods [147].

### 6.4 Modeling Continuous Structural Evolution

Real-world graphs, such as social networks, frequently undergo dynamic changes, with nodes and edges continuously evolving over time. While several studies have introduced flexible approaches to incorporate dynamic inputs during inference, either by adjusting the flow direction conditioned on new inputs or jumping in the vector field (see Section 4.1.1), the evolving dynamics of graph structures have received limited attention. Most existing methods either generate graphs using SDEs in a discrete manner [103] or adopt an end-to-end approach for graph generation tasks [72, 136, 148]. Although Huang et al. propose a framework that allows the dynamic evolution of both edge weights and node features, it overlooks the newly observed graph structures that emerge dynamically. Since graph topology significantly influences the evolution of node embeddings, it is crucial to incorporate the dynamic evolution of graph structures during inference to improve downstream performance.

### 6.5 Modeling Hierarchical Graph Dynamics

Hierarchical or Multi-scale data is essential for capturing complex structures and long-range dependencies that arise at different levels of granularity. While extensive research has been conducted on developing multi-scale GNNs [149, 150], relatively few studies have explored their integration with neural differential equations. From a data perspective, multi-scale characteristics can manifest at both temporal and spatial levels. From a modeling perspective, multi-scale information can be incorporated at different stages, such as during encoding or the modeling of differential equations, which yields multiple latent trajectories at different levels. Notably, Wang et al. [151] capture multi-scale temporal information at the encoding stage before applying a graph-based ODE. However, the incorporation of multi-scale spatial modeling within the differential equation remains an open area of research under the framework of Graph NDEs.

## 7 Conclusion

In this survey, we present the first comprehensive overview of Graph Neural Differential Equations (Graph NDEs), beginning with the fundamental concepts of both GNNs and differential equations. We then introduce a structured taxonomy covering tasks, graph construction methods, and the roles of GNNs in various settings. Methodologically, we analyze existing literature through two primary perspectives: Temporal Dynamics Modeling and Spatial Dynamics Modeling, highlighting key challenges and potential solutions. Additionally, we discuss diverse applications of Graph NDEs and identify persistent research gaps, suggesting directions for future study. By detailing how GNNs can be integrated more naturally with differential equation frameworks, we believe this survey will serve as a catalyst for continued innovation in this rapidly developing field, inspiring both researchers and practitioners to advance the state of the art in Graph NDEs.

## Acknowledgment

This research was partially supported by the US National Science Foundation under Award Number 2319449 and Award Number 2312502, as well as the US National Institute of Diabetes and Digestive and Kidney Diseases of the US National Institutes of Health under Award Number K25DK135913.

## References

- [1] Andrew Keane, Bernd Krauskopf, and Claire M Postlethwaite. Climate models with delay differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(11), 2017.
- [2] Elizabeth E Holmes, Mark A Lewis, JE Banks, and RR Veit. Partial differential equations in ecology: spatial interactions and population dynamics. *Ecology*, 75(1):17–29, 1994.
- [3] Xiangfeng Yang, Yuhan Liu, and Gyei-Kark Park. Parameter estimation of uncertain differential equation with application to financial market. *Chaos, Solitons & Fractals*, 139:110026, 2020.
- [4] Ting Dang, Jing Han, Tong Xia, Erika Bondareva, Chloë Siegele-Brown, Jagmohan Chauhan, Andreas Grammenos, Dimitris Spathis, Pietro Cicuta, and Cecilia Masclo. Conditional neural ode processes for individual disease progression forecasting: a case study on covid-19. In *Proceedings of the 29th ACM SIGKDD Conference On Knowledge Discovery and Data Mining*, pages 3914–3925, 2023.
- [5] Xiaoda Wang, Yufi Zhao, Kaiqiao Han, Xiao Luo, Sanne van Rooij, Jennifer Stevens, Lifang He, Liang Zhan, Yizhou Sun, Wei Wang, and Carl Yang. Conditional neural ode for longitudinal parkinson's disease progression forecasting. In *Abstract in the Organization for Human Brain Mapping Annual Meeting*, 2025.
- [6] Yoshihiro Maki and Hideo Hirose. Infectious disease spread analysis using stochastic differential equations for sir model. In *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, pages 152–156. IEEE, 2013.
- [7] Zewen Liu, Guancheng Wan, B Aditya Prakash, Max SY Lau, and Wei Jin. A review of graph neural networks in epidemic modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6577–6587, 2024.
- [8] John C Butcher. Numerical methods for ordinary differential equations in the 20th century. *Journal of Computational and Applied Mathematics*, 125(1-2):1–29, 2000.
- [9] D Sloan, S Vandewalle, and E Süli. Partial differential equations. 2012.
- [10] Nicolaas G Van Kampen. Stochastic differential equations. *Physics reports*, 24(3):171–228, 1976.
- [11] Ljupčo Todorovski and Sašo Džeroski. Integrating domain knowledge in equation discovery. In *Computational Discovery of Scientific Knowledge: Introduction, Techniques, and Applications in Environmental and Life Sciences*, pages 69–97. Springer, 2007.
- [12] Andrey Bryukhin, Jiahao Huang, Zhongying Deng, Guang Yang, Carola-Bibiane Schönlieb, and Angelica Aviles-Rivero. HAMLET: Graph Transformer Neural Operator for Partial Differential Equations, October 2024. arXiv:2402.03541 [cs].
- [13] Yash Kumar and Souvik Chakraborty. GrADE: A graph based data-driven solver for time-dependent nonlinear partial differential equations, August 2021. arXiv:2108.10639 [stat].
- [14] Hoyun Choi, Sungyeop Lee, B. Kahng, and Junghyo Jo. GNRK: Graph Neural Runge-Kutta method for solving partial differential equations, October 2023. arXiv:2310.00618 [cs].
- [15] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [16] Xiaomei Wang, Qi An, Zilong He, and Wei Fang. A literature review of social network analysis in epidemic prevention and control. *Complexity*, 2021(1):3816221, 2021.
- [17] Boris Medina-Salgado, Eddy Sánchez-DelaCruz, Pilar Pozos-Parra, and Javier E Sierra. Urban traffic flow prediction techniques: A review. *Sustainable Computing: Informatics and Systems*, 35:100739, 2022.
- [18] Guancheng Wan, Zijie Huang, Wanjia Zhao, Xiao Luo, Yizhou Sun, and Wei Wang. Rethink graphode generalization within coupled dynamical system. In *Forty-second International Conference on Machine Learning*, 2025.
- [19] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.
- [20] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. Pmlr, 2019.
- [21] Yezi Liu and Yanning Shen. Tinygraph: joint feature and node condensation for graph neural networks. *arXiv preprint arXiv:2407.08064*, 2024.
- [22] Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.
- [23] Louis-Pascal Khonneux, Meng Qu, and Jian Tang. Continuous graph neural networks. In *International conference on machine learning*, pages 10432–10441. PMLR, 2020.
- [24] Haoteng Tang, Guodong Liu, Siyuan Dai, Kai Ye, Kun Zhao, Wenlu Wang, Carl Yang, Lifang He, Alex Leow, Paul Thompson, et al. Interpretable spatio-temporal embedding for brain structural-effective network with ordinary differential equation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 227–237. Springer, 2024.
- [25] Andrey Bryukhin, Jiahao Huang, Zhongying Deng, Guang Yang, Carola-Bibiane Schönlieb, and Angelica Aviles-Rivero. Hamlet: Graph transformer neural operator for partial differential equations. *arXiv preprint arXiv:2402.03541*, 2024.
- [26] Suresh Bishnoi, Jayadeva Jayadeva, Sayan Ranu, and NM Anoop Krishnan. Brognet: Momentum-conserving graph neural stochastic differential equation for learning brownian dynamics. In *The Twelfth International Conference on Learning Representations*, 2024.
- [27] Hao Niu, Yuxiang Zhou, Xiaohao Yan, Jun Wu, Yuncheng Shen, Zhang Yi, and Junjie Hu. On the applications of neural ordinary differential equations in medical image analysis. *Artificial Intelligence Review*, 57(9):236, 2024.
- [28] Idris Bachali Losada and Nadia Terranova. Bridging pharmacology and neural networks: A deep dive into neural ordinary differential equations. *CPT: Pharmacometrics & Systems Pharmacology*, 13(8):1289–1296, 2024.
- [29] Sourabh Kumar Dubey, Hibah Islahi, and Raghvendra Singh. Deep neural networks for solving ordinary differential equations: A comprehensive review.
- [30] Andi Han, Dai Shi, Lequan Lin, and Junbin Gao. From continuous dynamics to graph neural networks: Neural diffusion and beyond. *arXiv preprint arXiv:2310.10121*, 2023.
- [31] Farzan Soleimani, Eric Paquet, Herna Lydia Viktor, and Wojtek Michalowski. Structure-based protein and small molecule generation using egnn and diffusion models: A comprehensive review. *Computational and Structural Biotechnology Journal*, 2024.
- [32] Yu Xie, Yanfeng Liang, Maoguo Gong, A Kai Qin, Yew-Soon Ong, and Tiantian He. Semisupervised graph neural networks for graph classification. *IEEE Transactions on Cybernetics*, 53(10):6222–6235, 2022.
- [33] Filipe De Avila Belbute-Peres, Thomas Economou, and Zico Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction. In *international conference on machine learning*, pages 2402–2411. PMLR, 2020.
- [34] Richard Bergna, Sergio Calvo-Ordoñez, Felix L. Opolka, Pietro Liò, and Jose Miguel Hernandez-Lobato. Uncertainty Modeling in Graph Neural Networks via Stochastic Differential Equations, September 2024. arXiv:2408.16115 [cs].
- [35] Shengchao Liu, Weitao Du, Zhi-Ming Ma, Hongyu Guo, and Jian Tang. A group symmetric stochastic differential equation model for molecule multimodal pretraining. In *International Conference on Machine Learning*, pages 21497–21526. PMLR, 2023.
- [36] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.
- [37] Yanfu Zhang, Shangqian Gao, Jian Pei, and Heng Huang. Improving Social Network Embedding via New Second-Order Continuous Graph Neural Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2515–2523, Washington DC USA, August 2022. ACM.
- [38] Chuyi Huang. Str-godes: Spatial-temporal-ridership graph odes for metro ridership prediction. *arXiv preprint arXiv:2107.04980*, 2021.
- [39] Jiayan Guo, Peiyian Zhang, Chaozhuo Li, Xing Xie, Yan Zhang, and Sunghun Kim. Evolutionary Preference Learning via Graph Nested GRU ODE for Session-based Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 624–634, October 2022. arXiv:2206.12779 [cs].
- [40] Lijing Wang, Aniruddha Adiga, Jiangzhuo Chen, Adam Sadilek, Srinivasan Venkatraman, and Madhav Marathe. CausalGNN: Causal-Based Graph Neural Networks for Spatio-Temporal Epidemic Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12191–12199, June 2022.
- [41] Zijie Huang, Yizhou Sun, and Wei Wang. Coupled graph ode for learning interacting system dynamics. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 705–715, 2021.
- [42] Zijie Huang, Wanjia Zhao, Jingdong Gao, Ziniu Hu, Xiao Luo, Yadi Cao, Yuanzhou Chen, Yizhou Sun, and Wei Wang. TANGO: Time-Reversal Latent GraphODE for Multi-Agent Dynamical Systems, October 2023. arXiv:2310.06427 [cs].
- [43] Peixiao Wang, Tong Zhang, Hengcai Zhang, Shifen Cheng, and Wangshu Wang. Adding attention to the neural ordinary differential equation for spatio-temporal prediction. *International Journal of Geographical Information Science*, 38(1):156–181, January 2024.
- [44] Ke Xu, Weizhi Zhang, Yuanjie Zhu, Zihe Song, and S Yu Philip. Node-sat: Temporal graph learning with neural ode-guided self-attention.
- [45] Ni Xiong, Yan Yang, Yongquan Jiang, and Xiaocao Ouyang. Diffusion Graph Neural Ordinary Differential Equation Network for Traffic Prediction. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, June 2023.

- ISSN: 2161-4407.
- [46] Xu Yan, Xiaoliang Fan, Peizhen Yang, Zonghan Wu, Shirui Pan, Longbiao Chen, Yu Zang, and Cheng Wang. ConTIG: Continuous Representation Learning on Temporal Interaction Graphs, September 2021. arXiv:2110.06088 [cs].
- [47] Chengxi Zang and Fei Wang. Neural Dynamics on Complex Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 892–902, Virtual Event CA USA, August 2020. ACM.
- [48] Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. *Advances in Neural Information Processing Systems*, 33:16177–16187, 2020.
- [49] Shaan Desai, Marios Mattheakis, and Stephen Roberts. Variational Integrator Graph Networks for Learning Energy Conserving Dynamical Systems. *Physical Review E*, 104(3):035310, September 2021. arXiv:2004.13688 [cs].
- [50] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 364–373, August 2021. arXiv:2106.12931 [cs].
- [51] Valerii Iakovlev, Markus Heinonen, and Harri Lähdesmäki. Learning continuous-time PDEs from sparse data with graph neural networks, January 2021. arXiv:2006.08956 [cs].
- [52] T. Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. Graph-Coupled Oscillator Networks. In *Proceedings of the 39th International Conference on Machine Learning*, pages 18888–18909. PMLR, June 2022. ISSN: 2640-3498.
- [53] Xiao Luo, Jingyang Yuan, Zijie Huang, Huiyu Jiang, Yifang Qin, Wei Ju, Ming Zhang, and Yizhou Sun. HOPE: High-order Graph ODE For Modeling Interacting Dynamics. In *Proceedings of the 40th International Conference on Machine Learning*, pages 23124–23139. PMLR, July 2023. ISSN: 2640-3498.
- [54] Kenta Niwa, Naonori Ueda, Hiroshi Sawada, Akinori Fujino, Shoichiro Takeda, Guoqiang Zhang, and W. Bastiaan Kleijn. CoordiNet: Constrained Dynamics Learning for State Coordination Over Graph. *IEEE Transactions on Signal and Information Processing over Networks*, 9:242–257, 2023. Conference Name: IEEE Transactions on Signal and Information Processing over Networks.
- [55] Song Wen, Hao Wang, Di Liu, Qilong Zhangli, and Dimitris Metaxas. Second-Order Graph ODEs for Multi-Agent Trajectory Forecasting. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5079–5088, Waikoloa, HI, USA, January 2024. IEEE.
- [56] Yang Liu, Jiaoshun Cheng, Haihong Zhao, Tingyang Xu, Peilin Zhao, Fuge Tsung, Jia Li, and Yu Rong. SEGNO: Generalizing Equivariant Graph Neural Networks with Physical Inductive Biases, March 2024. arXiv:2308.13212 [cs].
- [57] Moshe Eliasof, Eldad Haber, Eran Treister, and Carola-Bibiane B. Schönlieb. On The Temporal Domain of Differential Equation Inspired Graph Neural Networks. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR, April 2024. ISSN: 2640-3498.
- [58] Liyi Huang, Bowen Pang, Qiming Yang, Xiangnan Feng, and Wei Wei. Link prediction by continuous spatiotemporal representation via neural differential equations. *Knowledge-Based Systems*, 292:11161, May 2024.
- [59] Penglei Gao, Xi Yang, Rui Zhang, Ping Guo, John Y. Goulermas, and Kaizhu Huang. EgPDE-Net: Building Continuous Neural Networks for Time Series Prediction with Exogenous Variables, September 2023. arXiv:2208.01913 [cs].
- [60] Chrysoula Kosma, Giannis Nikolentzos, George Panagopoulos, Jean-Marc Steyaert, and Michalis Vazirgiannis. Neural Ordinary Differential Equations for Modeling Epidemic Spreading.
- [61] Masahito Uwamichi, Simon K. Schnyder, Tetsuya J. Kobayashi, and Satoshi Sawai. Integrating GNN and Neural ODEs for Estimating Non-Reciprocal Two-Body Interactions in Mixed-Species Collective Motion, November 2024. arXiv:2405.16503 [physics].
- [62] Jingyang Yuan, Gongbo Sun, Zhiping Xiao, Hang Zhou, Xiao Luo, Junyu Luo, Yusheng Zhao, Wei Ju, and Ming Zhang. Egode: An event-attended graph ode framework for modeling rigid dynamics. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [63] Jeongwhan Choi, Jinsung Jeon, and Noseong Park. Lt-ofc: Learnable-time ode-based collaborative filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 251–260, 2021.
- [64] Yuxuan Yang, Siyan Zhou, He Wang, Dongjing Wang, Xin Zhang, Dongjin Yu, and Shuguang Deng. Siamese learning based on graph differential equation for next-poi recommendation. *Applied Soft Computing*, 150:111086, 2024.
- [65] Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 8352–8364, 2021.
- [66] Yuelin Wang, Kai Yi, Xinliang Liu, Yu Guang Wang, and Shi Jin. ACMP: Allen-Cahn Message Passing for Graph Neural Networks with Particle Phase Transition, April 2023. arXiv:2206.05437 [cs].
- [67] Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. Learning Neural Ordinary Equations for Forecasting Future Links on Temporal Knowledge Graphs. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [68] Jiajia Wu and Ling Chen. Continuously Evolving Graph Neural Controlled Differential Equations for Traffic Forecasting, January 2024. arXiv:2401.14695 [cs].
- [69] Wenming Ma, Zihao Chu, Hao Chen, and Mingqi Li. Spatio-temporal evolutionary graph neural network for traffic flow prediction in UAV-based urban traffic monitoring system. *Scientific Reports*, 14(1):26800, November 2024.
- [70] Weiheng Zhong, Hadi Meidani, and Jane Macfarlane. Attention-based Spatial-Temporal Graph Neural ODE for Traffic Prediction, May 2023. arXiv:2305.00985 [cs].
- [71] Conditional Diffusion Based on Discrete Graph Structures for Molecular Graph Generation | Proceedings of the AAAI Conference on Artificial Intelligence.
- [72] Han Huang, Leilei Sun, Bowen Du, Yanjie Fu, and Weifeng Lv. GraphGDP: Generative Diffusion Processes for Permutation Invariant Graph Generation, December 2022. arXiv:2212.01842 [cs].
- [73] Konstantin Klemmer, Nathan S Safir, and Daniel B Neill. Positional encoder graph neural networks for geographic data. In *International conference on artificial intelligence and statistics*, pages 1379–1389. PMLR, 2023.
- [74] Wenxuan Ma, Shuang Li, Jingxuan Kang, et al. Language semantic graph guided data-efficient learning. *Advances in Neural Information Processing Systems*, 36:24088–24102, 2023.
- [75] Tom Z. Jiahao, Lishuo Pan, and M. Ani Hsieh. Learning to Swarm with Knowledge-Based Neural Ordinary Differential Equations, December 2021. arXiv:2109.04927 [cs].
- [76] Ben Chamberlain, James Rowbottom, Maria I. Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph Neural Diffusion. In *Proceedings of the 38th International Conference on Machine Learning*, pages 1407–1418. PMLR, July 2021. ISSN: 2640-3498.
- [77] Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. Beltrami Flow and Neural Diffusion on Graphs. In *Advances in Neural Information Processing Systems*, volume 34, pages 1594–1609. Curran Associates, Inc., 2021.
- [78] Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. GREAD: Graph Neural Reaction-Diffusion Networks. In *Proceedings of the 40th International Conference on Machine Learning*, pages 5722–5747. PMLR, July 2023. ISSN: 2640-3498.
- [79] Yao Zhang, Yun Xiong, Dongsheng Li, Caihua Shan, Kan Ren, and Yangyong Zhu. CoPE: Modeling Continuous Propagation and Evolution on Interaction Graph. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pages 2627–2636, New York, NY, USA, 2021. Association for Computing Machinery.
- [80] Yucheng Xing, Jacqueline Wu, Yingru Liu, Xuewen Yang, and Xin Wang. AG-GDN: A Continuous Stochastic Predictive Model for Monitoring Sporadic Time Series on Graphs. In Biao Luo, Long Cheng, Zheng-Guang Wu, Hongyi Li, and Chaojie Li, editors, *Neural Information Processing*, pages 130–146, Singapore, 2024. Springer Nature.
- [81] Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate Time Series Forecasting with Dynamic Graph Neural ODEs. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9168–9180, September 2023. arXiv:2202.08408 [cs].
- [82] Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.
- [83] Zijie Huang, Yizhou Sun, and Wei Wang. Coupled graph ode for learning interacting system dynamics. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*, pages 705–715, 2021.
- [84] Zijie Huang, Yizhou Sun, and Wei Wang. Generalizing graph ode for learning complex system dynamics across environments. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*, 2023.
- [85] Qingqing Long, Zheng Fang, Chen Fang, Chong Chen, Pengfei Wang, and Yuanchun Zhou. Unveiling delay effects in traffic forecasting: a perspective from spatial-temporal delay differential equations. In *Proceedings of the ACM Web Conference 2024*, pages 1035–1044, 2024.
- [86] Jiahao Li, Huandong Wang, and Xinxin Chen. Physics-informed Neural ODE for Post-disaster Mobility Recovery. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1587–1598, Barcelona Spain, August 2024. ACM.
- [87] Zhenyu Han, Yanxin Xi, Tong Xia, Yu Liu, and Yong Li. Devil in the landscapes: Inferring epidemic exposure risks from street view imagery. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pages 1–4, 2023.
- [88] Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ode integrators. *arXiv preprint arXiv:1909.12790*, 2019.
- [89] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*,

- 2020.
- [90] Zienkiewicz Olgierd Cecil, Taylor Robert Leroy, Nithiarasu Perumal, and Zhu JZ. The finite element method. vol. 3, 1977.
- [91] Zichao Jiang, Junyang Jiang, Qinghe Yao, and Gengchao Yang. A neural network-based PDE solving algorithm with high precision. *Scientific Reports*, 13(1):4479, March 2023.
- [92] Tiammin Han and Yuhuan Han. Numerical solution for super large scale systems. *IEEE Access*, 1:537–544, 2013.
- [93] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.
- [94] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [95] Keyulu Xu, Mozhi Zhang, Stefanie Jegelka, and Kenji Kawaguchi. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In *International Conference on Machine Learning*, pages 11592–11602. PMLR, 2021.
- [96] Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International conference on machine learning*, pages 3154–3164. PMLR, 2020.
- [97] Matthew Thorpe, Hedi Xia, Tan Nguyen, Thomas Strohmer, Andrea L Bertozzi, and Stanley J Osher. GRAND++: GRAPH NEURAL DIFFUSION WITH A SOURCE TERM. 2022.
- [98] Sohir Maskey, Raffaele Paolino, Aras Bacho, and Gitta Kutyniok. A Fractional Graph Laplacian Approach to Oversmoothing.
- [99] Yu Hao, Xin Cao, Yufan Sheng, Yixiang Fang, and Wei Wang. Ks-gnn: Keywords search over incomplete graphs via graphs neural network. *Advances in Neural Information Processing Systems*, 34:1700–1712, 2021.
- [100] Qitian Wu, Fan Nie, Chenxiao Yang, Tianyi Bao, and Junchi Yan. Graph out-of-distribution generalization via causal intervention. In *Proceedings of the ACM Web Conference 2024*, pages 850–860, 2024.
- [101] Richard Bergna, Felix Opolka, Pietro Liò, and Jose Miguel Hernandez-Lobato. Graph Neural Stochastic Differential Equations, August 2023. arXiv:2308.12316 [cs].
- [102] Xixun Lin, Wenxiao Zhang, Fengzhao Shi, Chuan Zhou, Lixin Zou, Xiangyu Zhao, Dawei Yin, Shirui Pan, and Yanan Cao. Graph neural stochastic diffusion for estimating uncertainty in node classification. In *Forty-first International Conference on Machine Learning*, 2024.
- [103] Guojun Liang, Prayag Tiwari, Slawomir Nowaczyk, Stefan Byttner, and Fernando Alonso-Fernandez. Dynamic causal explanation based diffusion-variational graph neural network for spatiotemporal forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
- [104] Yang Song, Qiyu Kang, Sijie Wang, Kai Zhao, and Wee Peng Tay. On the robustness of graph neural diffusion to topology perturbations. *Advances in Neural Information Processing Systems*, 35:6384–6396, 2022.
- [105] Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. Adversarial robustness in graph neural networks: A hamiltonian approach. *Advances in Neural Information Processing Systems*, 36, 2024.
- [106] Qiyu Kang, Kai Zhao, Yang Song, Yihang Xie, Yanan Zhao, Sijie Wang, Rui She, and Wee Peng Tay. Coupling graph neural networks with fractional order continuous dynamics: A robustness study. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13049–13058, 2024.
- [107] Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. Graph neural convection-diffusion with heterophily. *arXiv preprint arXiv:2305.16780*, 2023.
- [108] Acong Zhang and Ping Li. Unleashing the power of high-pass filtering in continuous graph neural networks. In *Asian Conference on Machine Learning*, pages 1683–1698. PMLR, 2024.
- [109] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.
- [110] Chengzhi Yao, Zhi Li, and Junbo Wang. Spatio-Temporal Hypergraph Neural ODE Network for Traffic Forecasting. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1499–1504, December 2023. ISSN: 2374-8486.
- [111] Jielong Yan, Yifan Feng, Shihui Ying, and Yue Gao. HYPERGRAPH DYNAMIC SYSTEM. 2024.
- [112] Guangsi Shi, Daokun Zhang, Ming Jin, Shirui Pan, and S Yu Philip. Towards complex dynamic physics system simulation with graph neural ordinary equations. *Neural Networks*, 176:106341, 2024.
- [113] Zijie Huang, Wanjia Zhao, Jingdong Gao, Ziniu Hu, Xiao Luo, Yadi Cao, Yuanzhou Chen, Yizhou Sun, and Wei Wang. Physics-informed regularization for domain-agnostic dynamical system modeling. *arXiv preprint arXiv:2410.06366*, 2024.
- [114] Hao Wu, Changhu Wang, Fan Xu, Jinbao Xue, Chong Chen, Xian-Sheng Hua, and Xiao Luo. Pure: Prompt evolution with graph ode for out-of-distribution fluid dynamics modeling. *Advances in Neural Information Processing Systems*, 37:104965–104994, 2025.
- [115] Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ode integrators. In *NeurIPS*, 2019.
- [116] Guangsi Shi et al. Towards complex dynamic physics system simulation with graph neural odes. *arXiv preprint arXiv:2305.12334*, 2023.
- [117] Zheng Fang et al. Spatial-temporal graph ode networks for traffic flow forecasting. In *KDD*, 2021.
- [118] Yuqiao Su et al. Graph ode recurrent neural networks for traffic flow forecasting. In *ICECE*, 2022.
- [119] Weiheng Zhong et al. Attention-based spatial-temporal graph neural ode for traffic prediction. *arXiv preprint arXiv:2305.00985*, 2023.
- [120] Zibo Liu et al. Graph-based multi-ode neural networks for spatio-temporal traffic forecasting. *TMLR*, 2023.
- [121] Lin Bai et al. Adaptive correlation graph neural ordinary differential equation for traffic flow forecasting. *Engineering Letters*, 2024.
- [122] Aminu Da'u and Naomie Salim. Recommendation system based on deep learning methods: a systematic review and new directions. *Artificial Intelligence Review*, 53(4):2709–2748, 2020.
- [123] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009(1):421425, 2009.
- [124] Yifang Qin, Wei Ju, Hongjun Wu, Xiao Luo, and Ming Zhang. Learning Graph ODE for Continuous-Time Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3224–3236, July 2024. arXiv:2304.07042 [cs].
- [125] Yuxuan Yang, Siyuan Zhou, He Weng, Dongjing Wang, Xin Zhang, Dongjin Yu, and Shiguang Deng. Siamese learning based on graph differential equation for Next-POI recommendation. *Applied Soft Computing*, 150:111086, January 2024.
- [126] Wei Duan, Zongchen Fan, Peng Zhang, Gang Guo, and Xiaogang Qiu. Mathematical and computational approaches to epidemic modeling: a comprehensive review. *Frontiers of Computer Science*, 9:806–826, 2015.
- [127] Zewen Liu, Yunxiao Li, Mingyang Wei, Guancheng Wan, Max S.Y. Lau, and Wei Jin. EpiLearn: A python library for machine learning in epidemic modeling. In *epiDAMIK 2024: The 7th International Workshop on Epidemiology meets Data Mining and Knowledge Discovery at KDD 2024*, 2024.
- [128] Alexander Rodríguez, Harshavardhan Kamarthi, Pulak Agarwal, Javen Ho, Mira Patel, Suchet Sapre, and B Aditya Prakash. Data-centric epidemic forecasting: A survey. *arXiv preprint arXiv:2207.09370*, 2022.
- [129] Junyi Gao, Rakshit Sharma, Cheng Qian, Lucas M Glass, Jeffrey Spaeder, Justin Romberg, Jimeng Sun, and Cao Xiao. Stan: spatio-temporal attention network for pandemic prediction using real-world evidence. *Journal of the American Medical Informatics Association*, 28(4):733–743, 2021.
- [130] Qi Cao, Renhe Jiang, Chuang Yang, Zipei Fan, Xuan Song, and Ryosuke Shibasaki. Mepognn: Metapopulation epidemic forecasting with graph neural networks. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 453–468. Springer, 2022.
- [131] Guancheng Wan, Zewen Liu, Max S. Y. Lau, B. Aditya Prakash, and Wei Jin. Epidemiology-Aware Neural ODE with Continuous Disease Transmission Graph, November 2024. arXiv:2410.00049 [cs].
- [132] Yanqiao Zhu, Yuanqi Du, Yinkai Wang, Yichen Xu, Jieyi Zhang, Qiang Liu, and Shu Wu. A survey on deep graph generation: Methods and applications. In *Learning on Graphs Conference*, pages 47–1. PMLR, 2022.
- [133] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- [134] Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International conference on artificial intelligence and statistics*, pages 4474–4484. PMLR, 2020.
- [135] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International conference on machine learning*, pages 10362–10383. PMLR, 2022.
- [136] Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. Conditional diffusion based on discrete graph structures for molecular graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4302–4311, 2023.
- [137] Dimitrios Angelis, Filippos Sofos, and Theodoros E Karakasis. Artificial intelligence in physical sciences: Symbolic regression trends and perspectives. *Archives of Computational Methods in Engineering*, 30(6):3845–3865, 2023.
- [138] Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *International Conference on Machine Learning*, pages 936–945. Pmlr, 2021.
- [139] Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K Reddy. Llm-sr: Scientific equation discovery via programming with large language models. *arXiv preprint arXiv:2404.18400*, 2024.
- [140] Jianxiang Yan, Guanghui Song, Ying Li, Zhaoji Zhang, and Yuhaio Chi. Enhanced odma with channel code design and pattern collision resolution for unsourced multiple access. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 3201–3206. IEEE, 2024.
- [141] Linhao Luo, Gholamreza Haffari, and Shirui Pan. Graph sequential neural ode process for link prediction on dynamic and sparse graphs. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages

- 778–786, 2023.
- [142] Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (llms). *arXiv preprint arXiv:2310.04668*, 2023.
- [143] Mohammad Hashemi, Shengbo Gong, Juntong Ni, Wenqi Fan, B Aditya Prakash, and Wei Jin. A comprehensive survey on graph reduction: Sparsification, coarsening, and condensation. *arXiv preprint arXiv:2402.03358*, 2024.
- [144] Marc Finzi, Andres Potapczynski, Matthew Choptuik, and Andrew Gordon Wilson. A stable and scalable method for solving initial value pdes with neural networks. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.
- [145] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.
- [146] Alim Samat, Erzhu Li, Peijun Du, Sicong Liu, and Junshi Xia. Gpu-accelerated catboost-forest for hyperspectral image classification via parallelized mrmr ensemble subspace feature selection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:3200–3214, 2021.
- [147] Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.
- [148] Yogesh Verma, Samuel Kaski, Markus Heinonen, and Vikas Garg. Modular Flows: Differential Molecular Generation.
- [149] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.
- [150] Mingjie Qiu, Zhiyi Tan, and Bing-Kun Bao. Msgnn: Multi-scale spatio-temporal graph neural network for epidemic forecasting. *Data Mining and Knowledge Discovery*, 38(4):2348–2376, 2024.
- [151] Keyi Wang, Jichao Zhan, Qinghua Si, Yueying Li, and Youyong Kong. Dynamic multi-scale spatio-temporal graph ode for metro ridership prediction. In *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, volume 7, pages 1501–1509. IEEE, 2024.
- [152] Huimin Huang, Luodi Xie, Mingzhe Liu, Jiajun Lin, and Hong Shen. An embedding model for temporal knowledge graphs with long and irregular intervals. *Knowledge-Based Systems*, 296:111893, 2024.
- [153] Kaiyuan Cui, Xinyan Wang, Zicheng Zhang, and Weichen Zhao. Graph neural aggregation-diffusion with metastability. *arXiv preprint arXiv:2403.20221*, 2024.
- [154] Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International conference on machine learning*, pages 1407–1418. PMLR, 2021.
- [155] Yuntao Shou, Tao Meng, Wei Ai, and Keqin Li. Dynamic graph neural ordinary differential equation network for multi-modal emotion recognition in conversation. *arXiv preprint arXiv:2412.02935*, 2024.
- [156] Nan Yin, Mengzhu Wan, Li Shen, Hitesh Laxmidhand Patel, Baopu Li, Bin Gu, and Huan Xiong. Continuous spiking graph neural networks. *arXiv preprint arXiv:2404.01897*, 2024.
- [157] Yinfeng Xiang, Chao Li, Shibo He, and Jiming Chen. Agc-ode: Adaptive graph controlled neural ode for human mobility prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2024.
- [158] Jiajia Wu and Ling Chen. Continuously evolving graph neural controlled differential equations for traffic forecasting. *arXiv preprint arXiv:2401.14695*, 2024.
- [159] Guancheng Wan, Zewen Liu, Max SY Lau, B Aditya Prakash, and Wei Jin. Epidemiology-aware neural ode with continuous disease transmission graph. *arXiv preprint arXiv:2410.00049*, 2024.
- [160] James Koch, Pranab Roy Chowdhury, Heng Wan, Parin Bhaduri, Jim Yoon, Vivek Srikrishnan, and W Brent Daniel. Graph neural ordinary differential equations for coarse-grained socioeconomic dynamics. *arXiv preprint arXiv:2407.18108*, 2024.
- [161] Yue Sun, Chao Chen, Yuesheng Xu, Sihong Xie, Rick S Blum, and Parv Venkatasubramaniam. Incorporating domain differential equations into graph convolutional networks to lower generalization discrepancy. *arXiv preprint arXiv:2404.01217*, 2024.
- [162] Ding Wang, Wei Zhou, and Songlin Hu. Information diffusion prediction with graph neural ordinary differential equation network. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9699–9708, 2024.
- [163] Yifang Qin, Wei Ju, Hongjun Wu, Xiao Luo, and Ming Zhang. Learning graph ode for continuous-time sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3224–3236, 2024.
- [164] Xiao Luo, Yiyang Gu, Huiyu Jiang, Hang Zhou, Jinsheng Huang, Wei Ju, Zhiping Xiao, Ming Zhang, and Yizhou Sun. Pgode: Towards high-quality system dynamics modeling. *arXiv preprint arXiv:2311.06554*, 2023.
- [165] Hemant Kumawat, Biswadeep Chakraborty, and Saibal Mukhopadhyay. Stem-fold: Stochastic temporal manifold for multi-agent interactions in the presence of hidden agents. In *6th Annual Learning for Dynamics & Control Conference*, pages 1427–1439. PMLR, 2024.
- [166] Lanlan Chen, Kai Wu, Jian Lou, and Jing Liu. Signed graph neural ordinary differential equation for modeling continuous-time dynamics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8292–8301, 2024.
- [167] Alessio Gravina, Daniele Zambon, Davide Bacciu, and Cesare Alippi. Temporal graph odes for irregularly-sampled time series. *arXiv preprint arXiv:2404.19508*, 2024.
- [168] Linzhi Li, Xiaofeng Zhou, Guoliang Hu, Shuai Li, and Dongni Jia. A recurrent spatio-temporal graph neural network based on latent time graph for multi-channel time series forecasting. *IEEE Signal Processing Letters*, 2024.
- [169] Haodong Cheng, Yingchi Mao, and Xiao Jia. A framework based on physics-informed graph neural ode: for continuous spatial-temporal pandemic prediction. *Applied Intelligence*, 54(24):12661–12675, 2024.
- [170] Zijie Huang, Jeehyun Hwang, Junkai Zhang, Jinwoo Baik, Weitong Zhang, Dominik Wodarz, Yizhou Sun, Quanquan Gu, and Wei Wang. Causal graph ode: Continuous treatment effect modeling in multi-agent dynamical systems. In *Proceedings of the ACM Web Conference 2024*, pages 4607–4617, 2024.
- [171] Kaiqiao Han, Yi Yang, Zijie Huang, Xuan Kan, Yang Yang, Ying Guo, Lifang He, Liang Zhan, Yizhou Sun, Wei Wang, et al. Brainode: Dynamic brain signal analysis via graph-aided neural ordinary differential equations. *arXiv preprint arXiv:2405.00077*, 2024.
- [172] Dhananjay Bhaskar, Daniel Sumner Magruder, Matheo Morales, Edward De Brouwer, Aarthi Venkat, Frederik Wenkel, Guy Wolf, and Smita Krishnaswamy. Inferring dynamic regulatory interaction graphs from time series data with perturbations. In *Learning on Graphs Conference*, pages 22–1. PMLR, 2024.
- [173] Song Wen, Hao Wang, Di Liu, Qilong Zhangli, and Dimitris Metaxas. Second-order graph odes for multi-agent trajectory forecasting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5101–5110, 2024.
- [174] Kenta Niwa, Naonori Ueda, Hiroshi Sawada, Akinori Fujino, Shoichiro Takeda, Guoqiang Zhang, and W Bastiaan Kleijn. Coordinet: Constrained dynamics learning for state coordination over graph. *IEEE Transactions on Signal and Information Processing over Networks*, 9:242–257, 2023.
- [175] Theodor Westny, Joel Oskarsson, Björn Olofsson, and Erik Frisk. Evaluation of differentially constrained motion models for graph-based trajectory prediction. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8. IEEE, 2023.
- [176] Theodor Westny, Joel Oskarsson, Björn Olofsson, and Erik Frisk. Mtp-go: Graph-based probabilistic multi-agent trajectory prediction with neural odes. *IEEE Transactions on Intelligent Vehicles*, 8(9):4223–4236, 2023.
- [177] Xiao Luo, Jingyang Yuan, Zijie Huang, Huiyu Jiang, Yifang Qin, Wei Ju, Ming Zhang, and Yizhou Sun. Hope: High-order graph ode for modeling interacting dynamics. In *International conference on machine learning*, pages 23124–23139. PMLR, 2023.
- [178] Tiexin Qin, Benjamin Walker, Terry Lyons, Hong Yan, and Haoliang Li. Learning dynamic graph embeddings with neural controlled differential equations. *arXiv preprint arXiv:2302.11354*, 2023.
- [179] Masahito Uwamichi, Simon K. Schnyder, Tetsuya J. Kobayashi, and Satoshi Sawai. Integrating GNN and neural ODEs for estimating non-reciprocal two-body interactions in mixed-species collective motion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [180] Moshe Eliasof, Eldad Haber, and Eran Treister. Feature transportation improves graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11874–11882, 2024.
- [181] Yanna Ding, Zijie Huang, Malik Magdon-Ismail, and Jianxi Gao. Predicting time series of networked dynamical systems without knowing topology. *arXiv preprint arXiv:2412.18734*, 2024.
- [182] Yucheng Xing and Xin Wang. Hdg-ode: A hierarchical continuous-time model for human pose forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14700–14712, 2023.
- [183] Zijie Huang, Wanjia Zhao, Jingdong Gao, Ziniu Hu, Xiao Luo, Yadi Cao, Yuanzhou Chen, Yizhou Sun, and Wei Wang. Tango: Time-reversal latent graphode for multi-agent dynamical systems. *arXiv preprint arXiv:2310.06427*, 2023.
- [184] Zihao Chu, Wenming Ma, Mingqi Li, and Hao Chen. Adaptive decision spatio-temporal neural ode for traffic flow forecasting with multi-kernel temporal dynamic dilation convolution. *Neural Networks*, 179:106549, 2024.
- [185] Weizhi Zhang, Liangwei Yang, Zihe Song, Henry Peng Zou, Ke Xu, Liancheng Fang, and Philip S Yu. Do we really need graph convolution during training? light post-training graph-ode for efficient recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 3248–3258, 2024.
- [186] Suresh Bishnoi, Ravinder Bhattoo, Sayan Ranu, and NM Krishnan. Enhancing the inductive biases of graph neural ode for modeling dynamical systems. *arXiv preprint arXiv:2209.10740*, 2022.
- [187] Omid Bazgir, Zichen Wang, Ji Won Park, Marc Hafner, and James Lu. Integration of graph neural network and neural-odes for tumor dynamic prediction. *arXiv preprint arXiv:2310.00926*, 2023.

- [188] Junkai Mao, Yuexing Han, and Bing Wang. Mpstan: Metapopulation-based spatio-temporal attention network for epidemic forecasting. *Entropy*, 26(4):278, 2024.
- [189] Haiquan Qiu, Shuzhi Liu, and Quanming Yao. Neural symbolic regression of complex network dynamics. *arXiv preprint arXiv:2410.11185*, 2024.
- [190] Zhaobin Mo, Yongjie Fu, and Xuan Di. Pi-neugode: Physics-informed graph neural ordinary differential equations for spatiotemporal trajectory prediction. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 1418–1426, 2024.
- [191] Ruikun Li, Huadong Wang, Jinghua Piao, Qingmin Liao, and Yong Li. Predicting long-term dynamics of complex networks via identifying skeleton in hyperbolic space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1655–1666, 2024.
- [192] Wenming Ma, Zihao Chu, Hao Chen, and Mingqi Li. Spatio-temporal environmental graph neural network for traffic flow prediction in uav-based urban traffic monitoring system. *Scientific Reports*, 14(1):26800, 2024.
- [193] Hao Wu, Changhu Wang, Fan Xu, Jinbao Xue, Chong Chen, Xian-Sheng Hua, and Xiao Luo. Pure: Prompt evolution with graph ode for out-of-distribution fluid dynamics modeling. *Advances in Neural Information Processing Systems*, 37:104965–104994, 2024.
- [194] Chrysoula Kosma, Giannis Nikolenzatos, George Panagopoulos, Jean-Marc Steyaert, and Michalis Vazirgiannis. Neural ordinary differential equations for modeling epidemic spreading. *Transactions on Machine Learning Research*, 2023.
- [195] Alessio Gravina, Davide Bacci, and Claudio Gallicchio. Anti-symmetric dgn: a stable architecture for deep graph networks. *arXiv preprint arXiv:2210.09789*, 2022.
- [196] Weihehng Zhong, Hadi Meidani, and Jane Macfarlane. Attention-based spatial-temporal graph neural ode for traffic prediction. *arXiv preprint arXiv:2305.00985*, 2023.
- [197] Song Jiang, Zijie Huang, Xiao Luo, and Yizhou Sun. Cf-gode: Continuous-time causal inference for multi-agent dynamical systems. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 997–1009, 2023.
- [198] Ni Xiong, Yan Yang, Yongquan Jiang, and Xiaocao Ouyang. Diffusion graph neural ordinary differential equation network for traffic prediction. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2023.
- [199] Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. Gread: Graph neural reaction-diffusion networks. In *International Conference on Machine Learning*, pages 5722–5747. PMLR, 2023.
- [200] Zibo Liu, Parshin Shojaee, and Chandan K Reddy. Graph-based multi-ode neural networks for spatio-temporal traffic forecasting. *arXiv preprint arXiv:2305.18687*, 2023.
- [201] Fei Zhan, Xiaofeng Zhou, Shuai Li, Dongni Jia, and Hong Song. Learning latent odes with graph rnn for multi-channel time series forecasting. *IEEE Signal Processing Letters*, 30:1432–1436, 2023.
- [202] Hui Zeng, Chaojie Jiang, Yuanchun Lan, Xiaohui Huang, Junyang Wang, and Xinhua Yuan. Long short-term fusion spatial-temporal graph convolutional networks for traffic flow forecasting. *Electronics*, 12(1):238, 2023.
- [203] Yue Sun, Chao Chen, Yuesheng Xu, Sihong Xie, Rick S Blum, and Parv Venkatasubramaniam. Reaction-diffusion graph ordinary differential equation networks: Traffic-law-informed speed prediction under mismatched data. The 12th International Workshop on Urban Computing, held in conjunction with ..., 2023.
- [204] Shuang Wang, Hong Dai, Lin Bai, Chengrui Liu, and Junhong Chen. Temporal branching-graph neural ode without prior structure for traffic flow forecasting. *Eng. Let.*, 31:1534, 2023.
- [205] Joel Oskarsson, Per Sidén, and Fredrik Lindsten. Temporal graph neural networks for irregular data. In *International Conference on Artificial Intelligence and Statistics*, pages 4515–4531. PMLR, 2023.
- [206] Yi Xie, Yun Xiong, Jiawei Zhang, Chao Chen, Yao Zhang, Jie Zhao, Yizhu Jiao, Jinjing Zhao, and Yangyong Zhu. Temporal super-resolution traffic flow forecasting via continuous-time network dynamics. *Knowledge and Information Systems*, 65(11):4687–4712, 2023.
- [207] Thomas Asikis, Lucas Böttcher, and Nino Antulov-Fantulin. Neural ordinary differential equation control of dynamics on graphs. *Physical Review Research*, 4(1):013221, 2022.
- [208] Song Wen, Hao Wang, and Dimitris Metaxas. Social ode: Multi-agent trajectory forecasting with neural ordinary differential equations. In *European Conference on Computer Vision*, pages 217–233. Springer, 2022.
- [209] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 364–373, 2021.
- [210] Shaan A Desai, Marios Mattheakis, and Stephen J Roberts. Variational integrator graph networks for learning energy-conserving dynamical systems. *Physical Review E*, 104(3):035310, 2021.
- [211] Giangiocomo Mercatali, Andre Freitas, and Jie Chen. Graph neural flows for unveiling systemic interactions among irregularly sampled time series. *arXiv preprint arXiv:2410.14030*, 2024.
- [212] Yang Liu, Jiashun Cheng, Haihong Zhao, Tingyang Xu, Peilin Zhao, Fugee Tsung, Jia Li, and Yu Rong. Segno: Generalizing equivariant graph neural networks with physical inductive biases. *arXiv preprint arXiv:2308.13212*, 2023.
- [213] Abishek Thangamuthu, Gunjan Kumar, Suresh Bishnoi, Ravinder Bhattoo, NM Krishnan, and Sayan Ranu. Unravelling the performance of physics-informed graph neural networks for dynamical systems. *Advances in Neural Information Processing Systems*, 35:3691–3702, 2022.
- [214] Yanna Ding, Zijie Huang, Xiao Shou, Yihang Guo, Yizhou Sun, and Jianxi Gao. Architecture-aware learning curve extrapolation via graph ordinary differential equation. *arXiv preprint arXiv:2412.15554*, 2024.
- [215] Fang Sun, Zijie Huang, Haixin Wang, Yadi Cao, Xiao Luo, Wei Wang, and Yizhou Sun. Graph fourier neural odes: Bridging spatial and temporal multiscales in molecular dynamics. *arXiv preprint arXiv:2411.01600*, 2024.
- [216] Zhe Cui, Di Zang, Hong Zhu, and Keshuang Tang. Predictive and multigranularity resilience assessment of urban transportation based on neural controlled differential equation. *IEEE Transactions on Reliability*, 2024.
- [217] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 6367–6374, 2022.
- [218] Richard Bergna, Sergio Calvo-Ordonez, Felix L Opalka, Pietro Liò, and Jose Miguel Hernandez-Lobato. Uncertainty modeling in graph neural networks via stochastic differential equations. *arXiv preprint arXiv:2408.16115*, 2024.
- [219] Huafeng Cao, Zhongbao Zhang, Li Sun, and Zhi Wang. Inductive and irregular dynamic network representation based on ordinary differential equations. *Knowledge-Based Systems*, 227:107271, September 2021.
- [220] Ming Jin, Yuan-Fang Li, and Shirui Pan. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. *Advances in Neural Information Processing Systems*, 35:19874–19886, 2022.
- [221] Jeongwhan Choi, Jinsung Jeon, and Noseong Park. LT-OCF: Learnable-Time ODE-based Collaborative Filtering, August 2021. *arXiv:2108.06208 [cs]*.
- [222] Zihui Wang, Peizhen Yang, Xiaoliang Fan, Xu Yan, Zonghan Wu, Shirui Pan, Longbiao Chen, Yu Zang, Cheng Wang, and Rongshan Yu. Contig: Continuous representation learning on temporal interaction graphs. *Neural Networks*, 172:106151, 2024.

# Appendices

## A Summary of Graph NDEs

**Table 1: Comprehensive Summary of Graph Neural Differential Equations.**

Task	Paper	Role of GNN	DE Type	Graph Construction
Node/Graph Classification	[22, 152]	DE	First Order ODE	Dynamic
	[23, 47, 66, 97, 98, 105, 108, 153–155]	DE	First Order ODE	Static
	[52]	DE	Second Order ODE	Static
	[106]	DE	Fractional Order ODE	Static
	[101]	Encoder	First Order SDE	Static
	[34, 102]	DE	First Order SDE	Static
	[111]	DE	First Order ODE	Static
	[77, 104, 107]	DE	First Order PDE	Static
	[156]	DE	First Order PDE & Second Order PDE	Static
	[80]	DE	First Order ODE & First Order SDE	Dynamic
Forecasting	[81, 86, 157–167]	DE	First Order ODE	Dynamic
	[83, 168–170]	Encoder & DE	First Order ODE	Dynamic
	[40, 171, 172]	Encoder	First Order ODE	Dynamic
	[173–175]	DE	Second Order ODE	Dynamic
	[176]	Encoder	Second Order ODE	Dynamic
	[177]	Encoder & DE	Second Order ODE	Dynamic
	[178]	DE	First-order NCDE (Neural Controlled Differential Equation)	Dynamic
	[103]	Encoder	First Order SDE	Dynamic
	[89]	DE	Second Order PDE	Dynamic
	[179]	DE	First Order ODE & First Order SDE	Static
	[13]	DE	Second Order PDE	Static
	[12]	Encoder	First Order PDE	Static
	[51]	DE	High Order PDE	Static
	[85, 180]	DE	First Order DDE	Static
	[48, 82, 84, 181–183]	Encoder & DE	First Order ODE	Static
	[38, 62, 110, 112, 184–211]	DE	First Order ODE	Static
	[212, 213]	DE	Second Order ODE	Static
Link Prediction	[214, 215]	Encoder	First Order ODE	Static
	[216, 217]	DE	First-order NCDE (Neural Controlled Differential Equation)	Static
Ranking	[26, 218]	DE	First Order SDE	Static
	[88]	DE	First Order PDE	Static
	[198]	DE & Decoder	First Order ODE	Static
Graph Generation	[219, 220]	DE	First Order ODE	Dynamic
	[44]	Decoder	First Order ODE	Dynamic
	[221]	DE	First Order ODE	Static
Graph Generation	[79]	DE	First Order ODE	Dynamic
	[222]	Decoder	First Order ODE	Dynamic
	[148]	DE	First Order PDE	Dynamic
Graph Generation	[72, 134, 135]	DE	First Order SDE	Dynamic
	[136]	Encoder	Third Order SDE	Dynamic